NASA'S MISSION TO PLANET EARTH

EARTH PROBES

DATA INFORMATION SYSTEM

EOS

EARTH OBSERVING SYSTEM

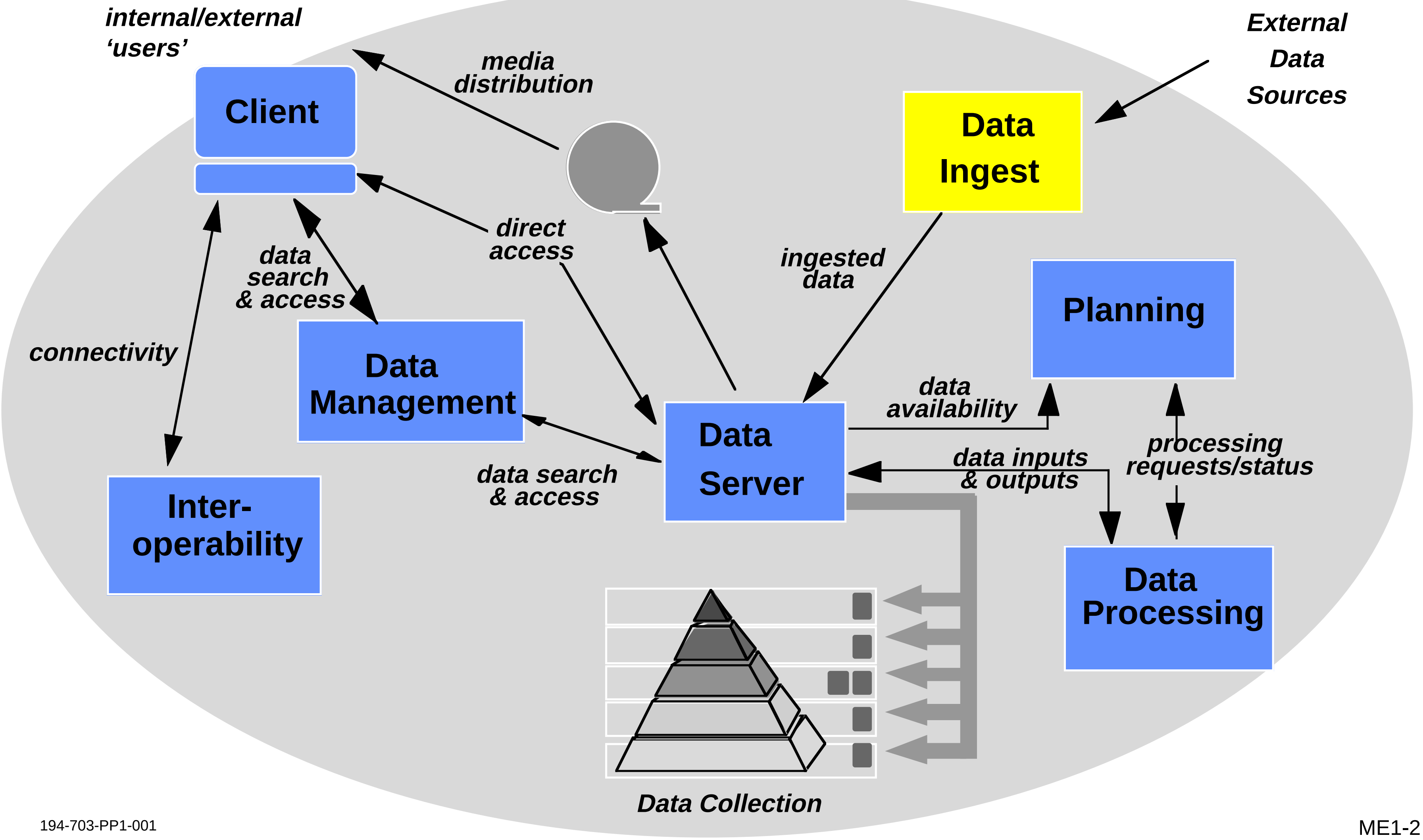
# Ingest, Planning & Processing Subsystems

## Mark Elkington

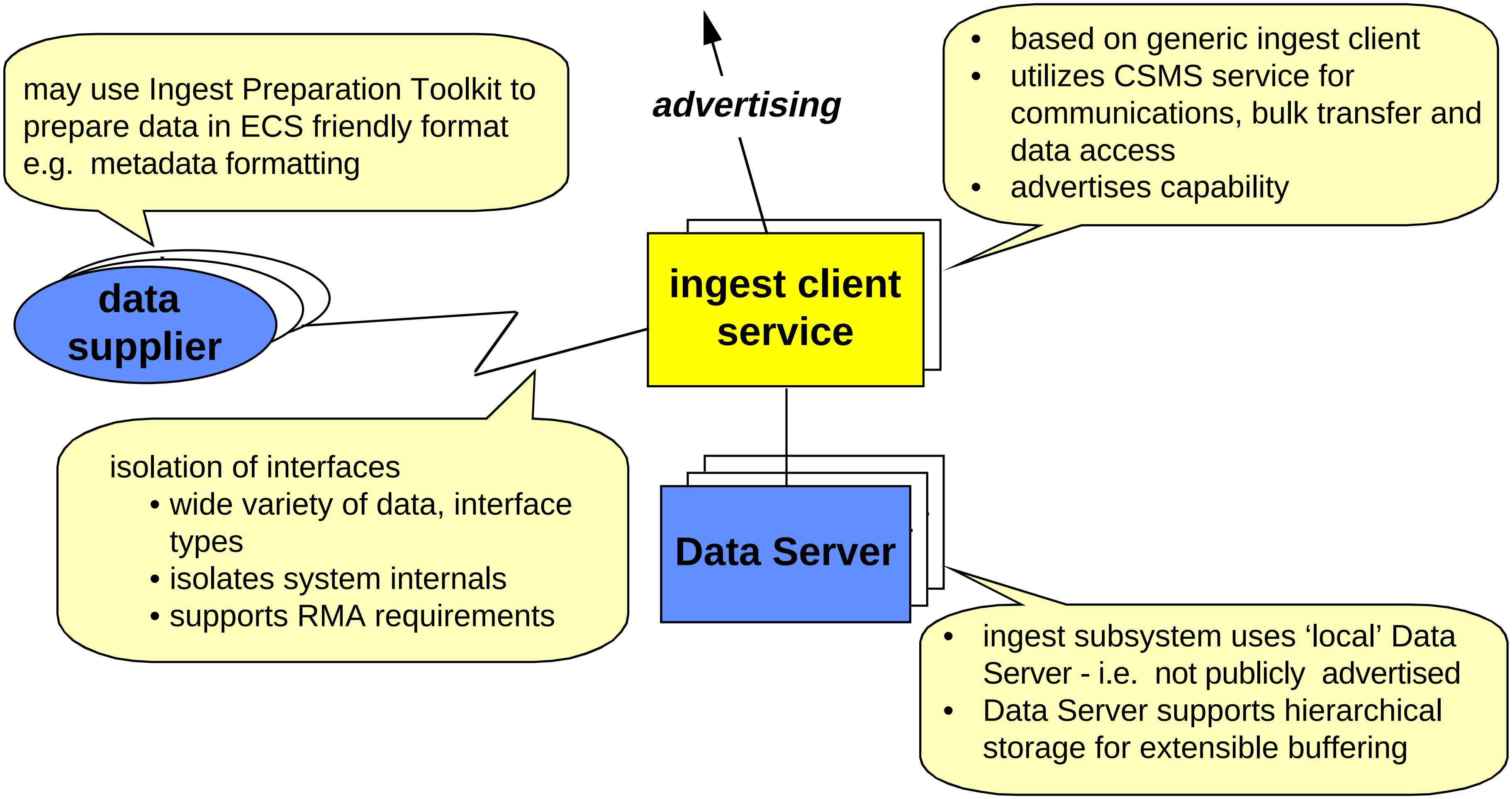
### System Design Review - 28 June 1994

SDS - 4.5.5

# Ingest Subsystem Context



# Ingest



advertising

may use Ingest Preparation Toolkit to prepare data in ECS friendly format e.g.  metadata formatting

data supplier

ingest client service

Data Server

- based on generic ingest client
- utilizes CSMS service for communications, bulk transfer and data access
- advertises capability

isolation of interfaces

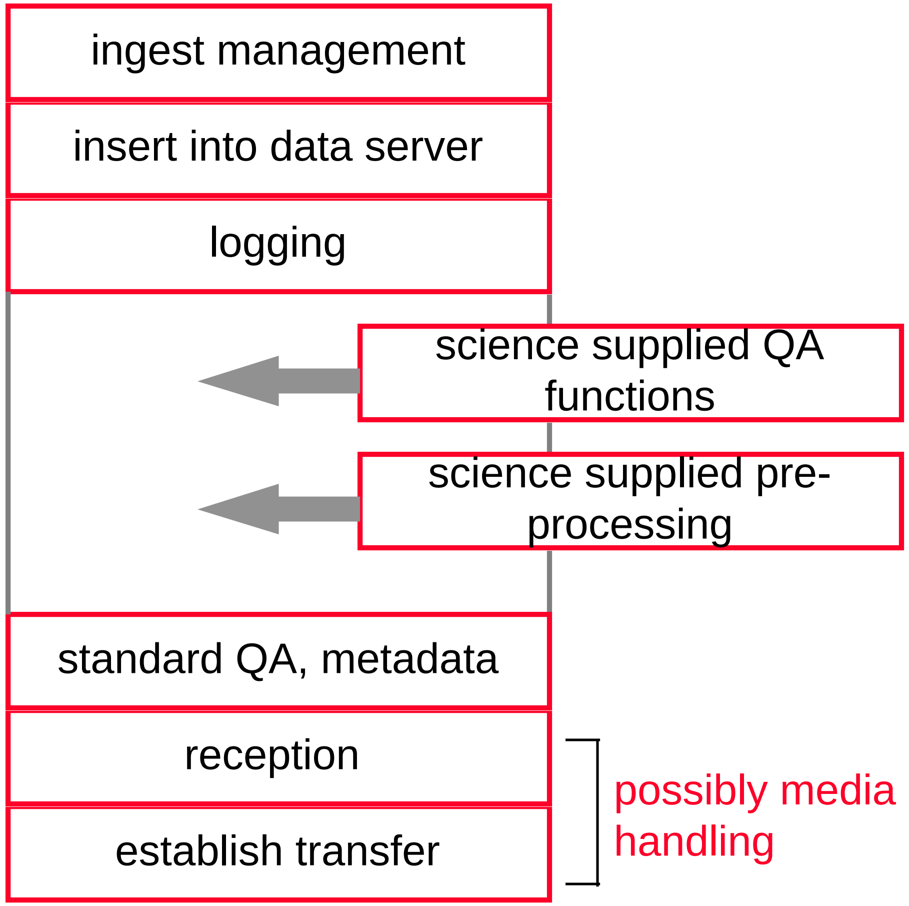
- wide variety of data, interface types
- isolates system internals
- supports RMA requirements

- ingest subsystem uses ‘local’ Data Server - i.e.  not publicly  advertised
- Data Server supports hierarchical storage for extensible buffering

SDS - 4.5.5

# Ingest

## Generic Ingest Client

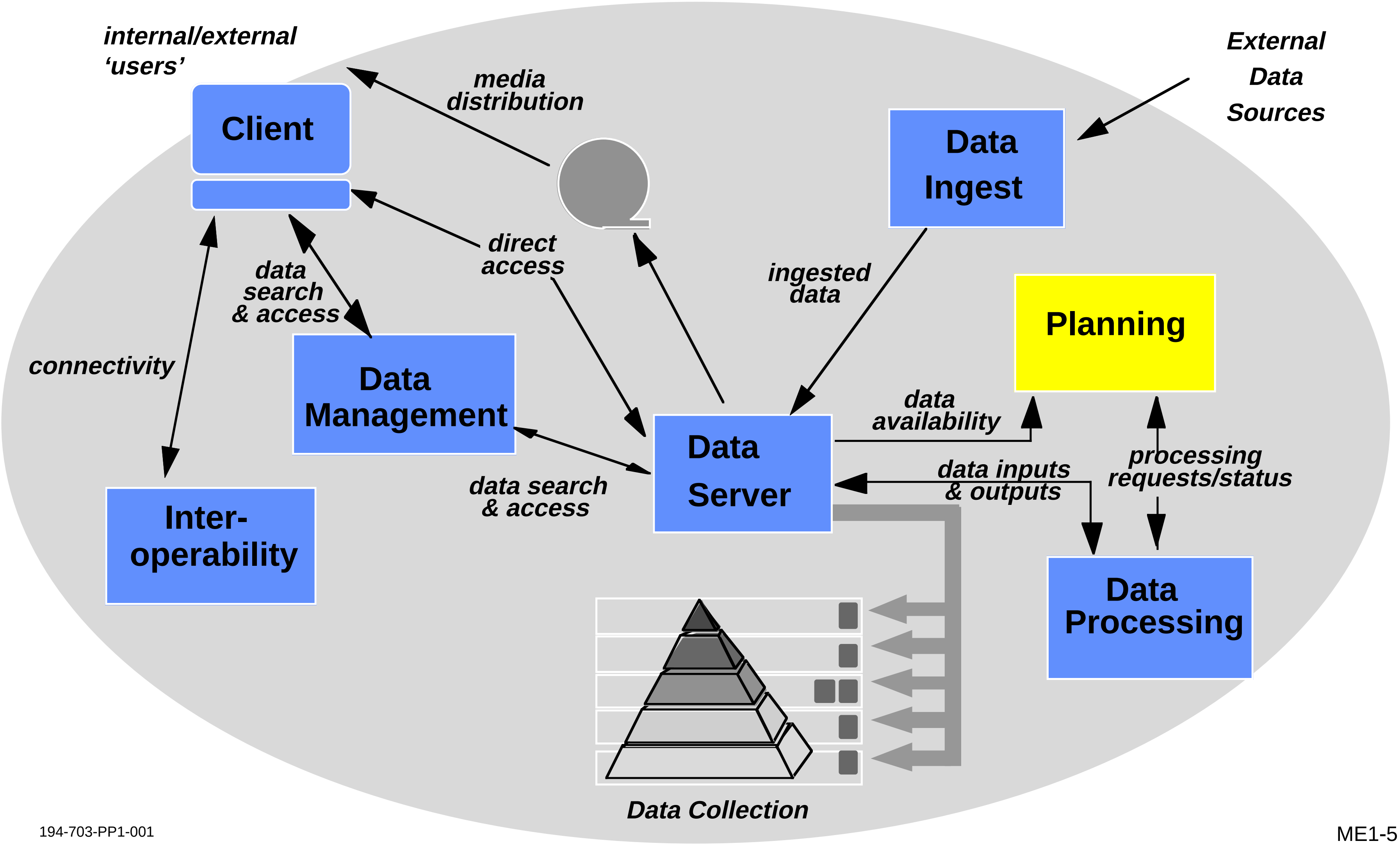


### Characteristics

- interface specializations
- can recognize collections
- supports predictive/non-predictive ingest
- logging
- problem resolution can be automatic, manual or combination
- allows insertion of science or DAAC specific functionality
- data preparation supported by distributable toolkit

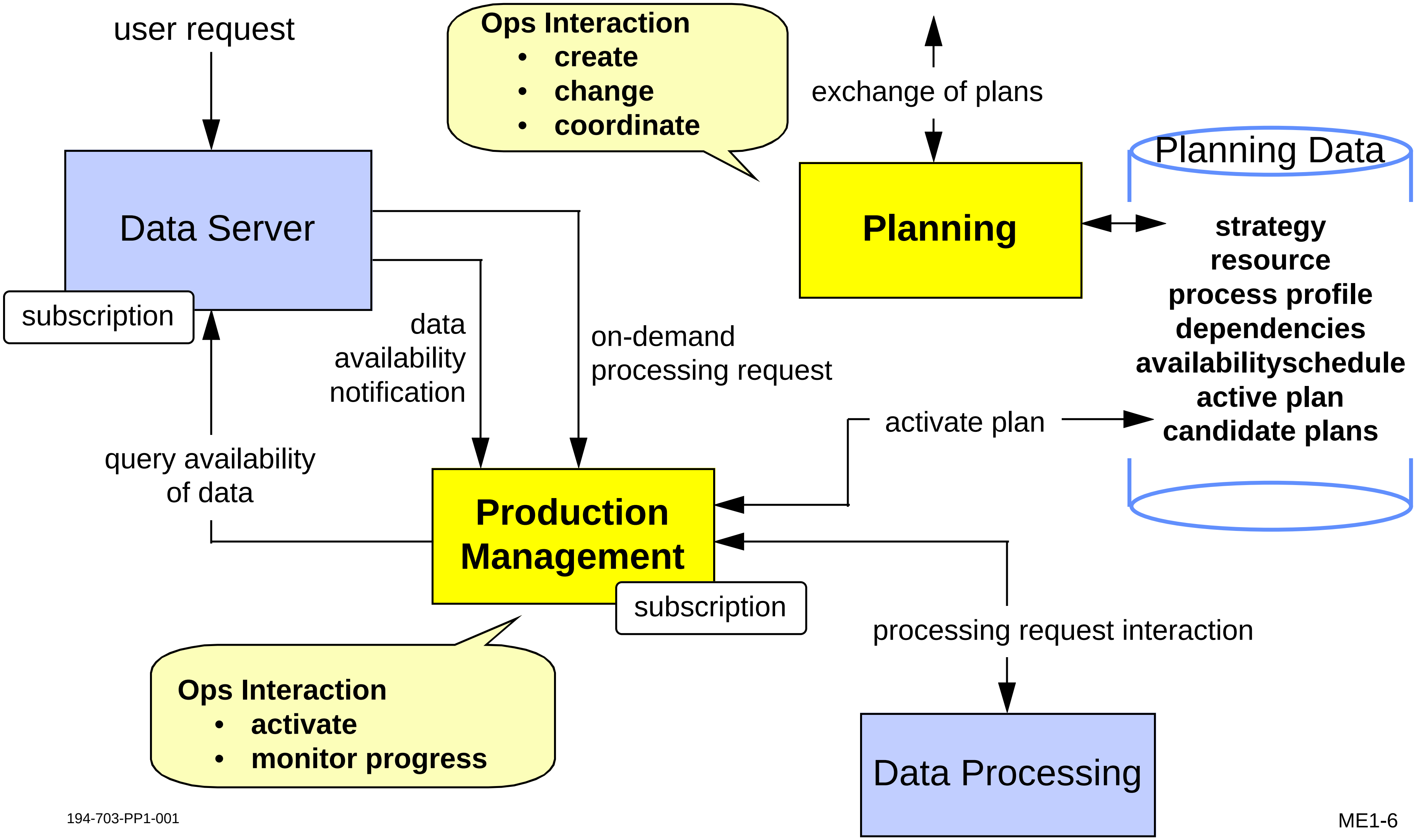
SDS - 4.5.6

# Planning Subsystem Context



SDS - 4.5.6

# Planning



user request

Data Server

subscription

data availability notification

on-demand processing request

query availability of data

Ops Interaction

- **create**
- **change**
- **coordinate**

exchange of plans

**Planning**

Planning Data

**strategy**
**resource**
**process profile**
**dependencies**
**availabilityschedule**
**active plan**
**candidate plans**

activate plan

**Production Management**

subscription

processing request interaction

Data Processing

**Ops Interaction**

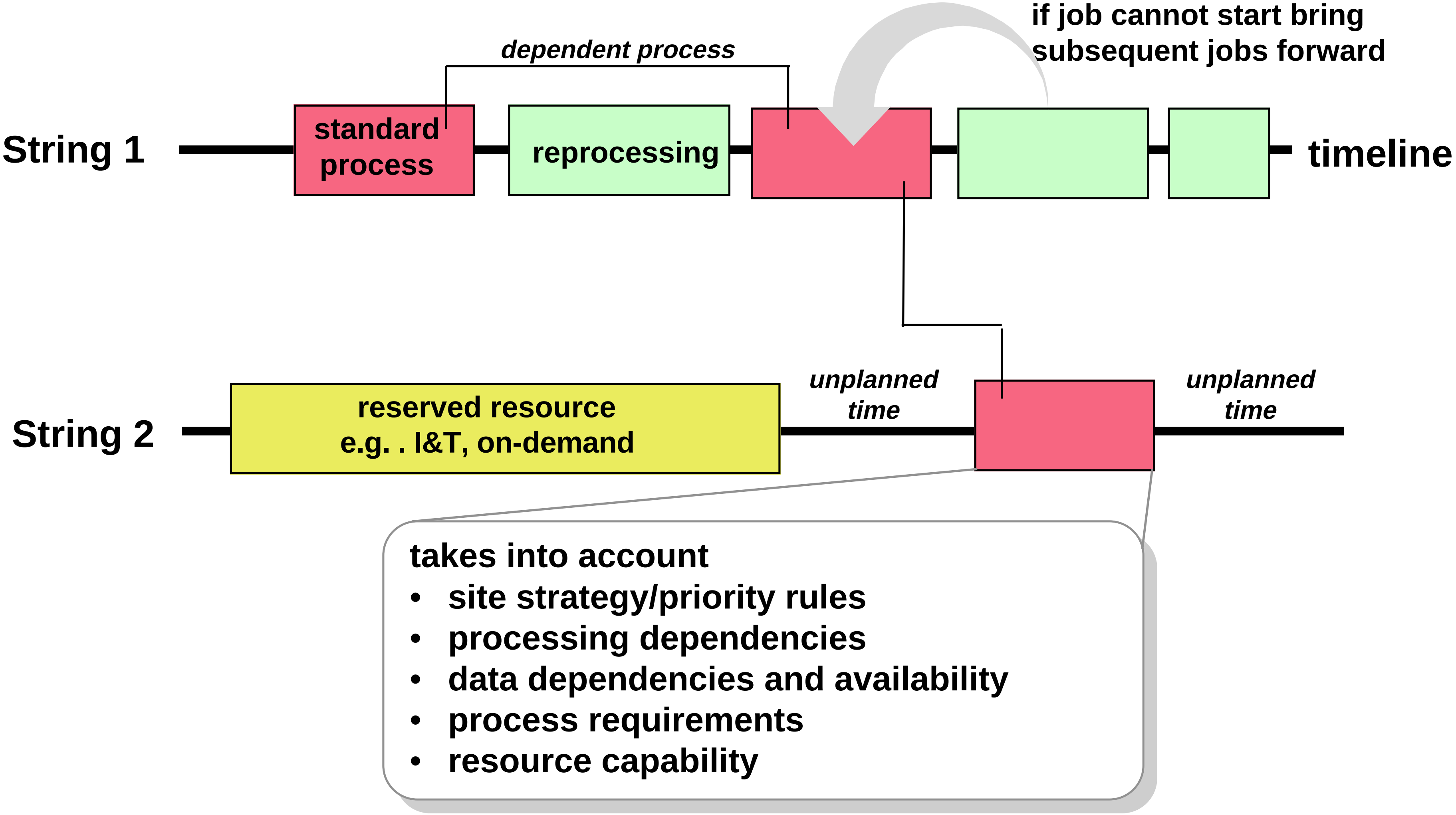
- **activate**
- **monitor progress**

# Planning

## Characteristics

- ‘places’ a *production strategy* onto the site resources dedicated to processing - uses PGE as executable unit
- recognizes *processing types, resource pools* - site definable .
- separated from processing to support future evolution
- supports multiple plans (candidate and active)
- supports exchange of plans and visualization of other site plans - also functionality at SMC
- supports subscriptions - for notification of changing plans
- supports both push and (some) pull processing - data driven and on-demand

# The Plan



takes into account

- site strategy/priority rules
- processing dependencies
- data dependencies and availability
- process requirements
- resource capability

# Planning

## Ongoing Design Studies

- production rules (complexity and extent)
- guaranteed processing time (is reduced optimization important?)
- multi-tasking (does it make sense?)

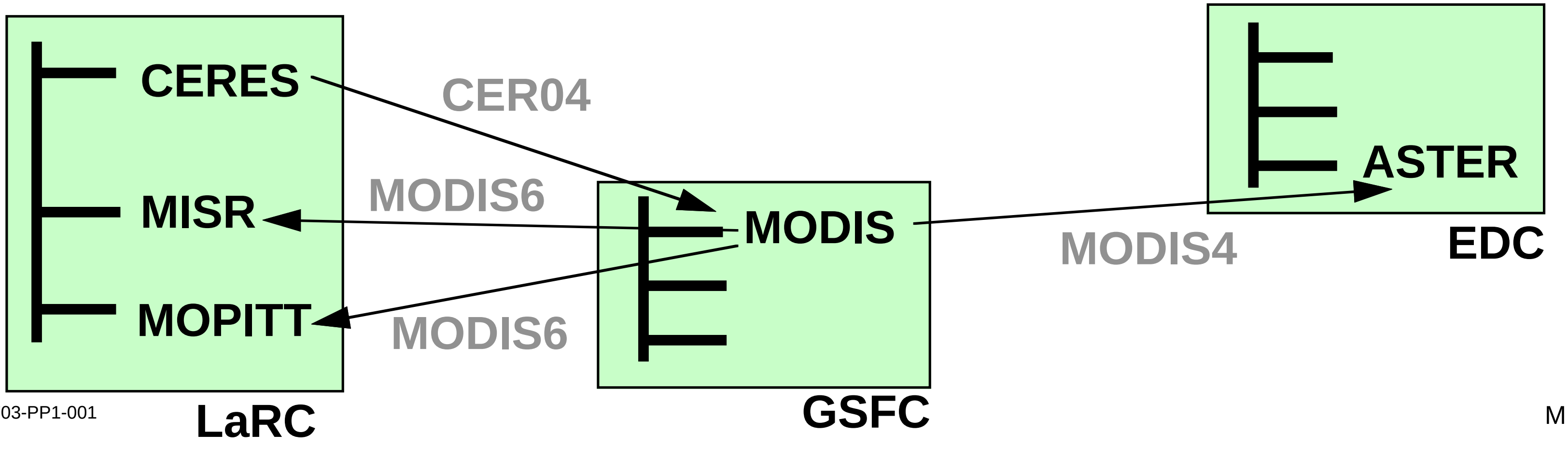
## Prototype Investigation

- investigation of several OTS packages and libraries
- customization of ‘Delphi’ object class libraries [cf FOS planning/scheduling]
- investigate outstanding issues related to planning  - verification with DAACs and instrument teams
- support definition of Level 4 requirements

# Inter-Site Planning

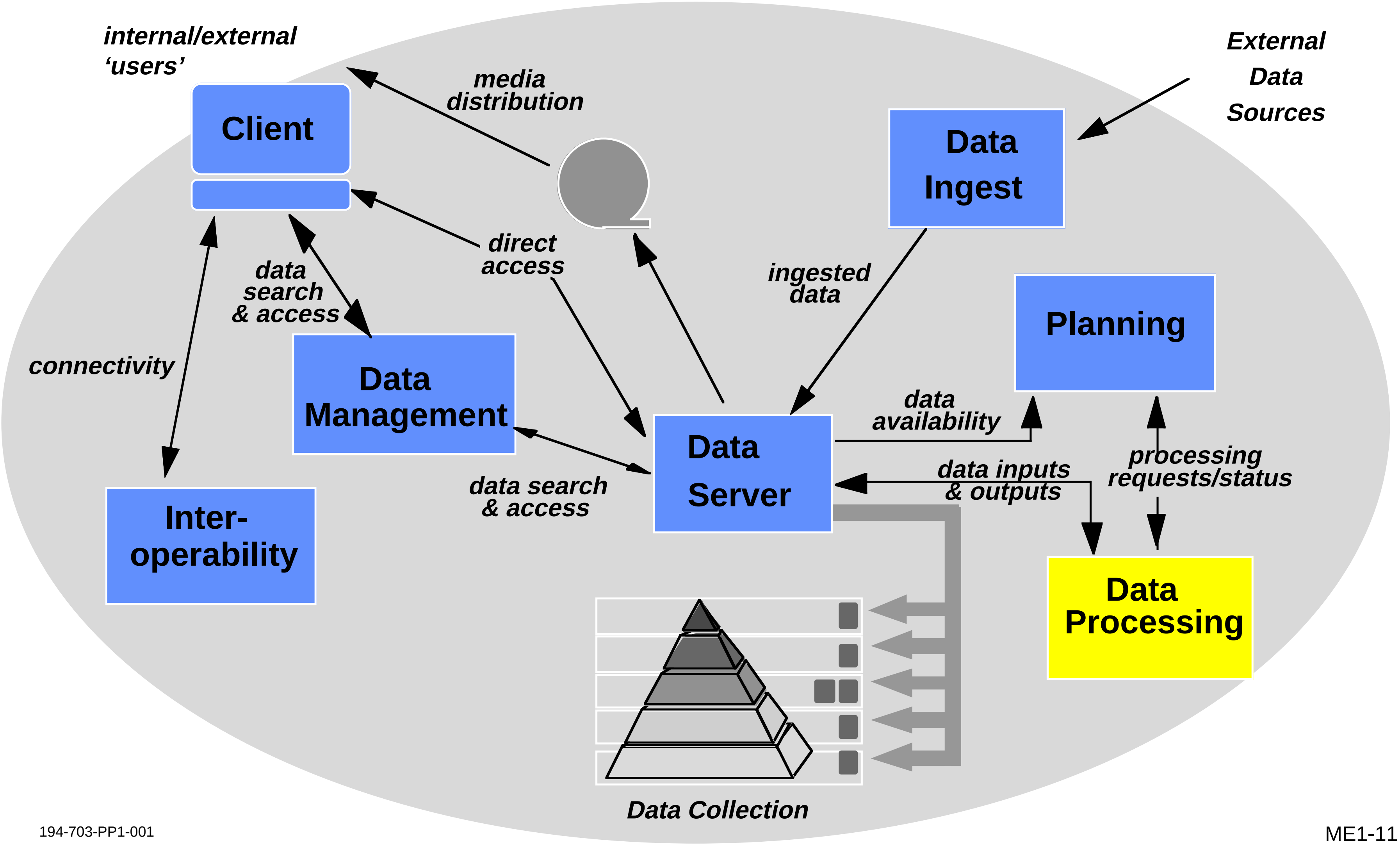
## Design supports a variety of policy approaches . . . .

- global viewing of product dependencies and schedules through combining of plans in planning application (at DAAC and/or SMC)
- DAACS can either:
  - collaborate independently
  - be free to change plans in predefined ways - otherwise use SMC to coordinate
  - always confirm changes through SMC
- Plans can be constructed to be resilient to temporary changes
- Subscriptions can be placed on key events at other DAACS (e.g. changes to active plan, ingest client, resource availability)



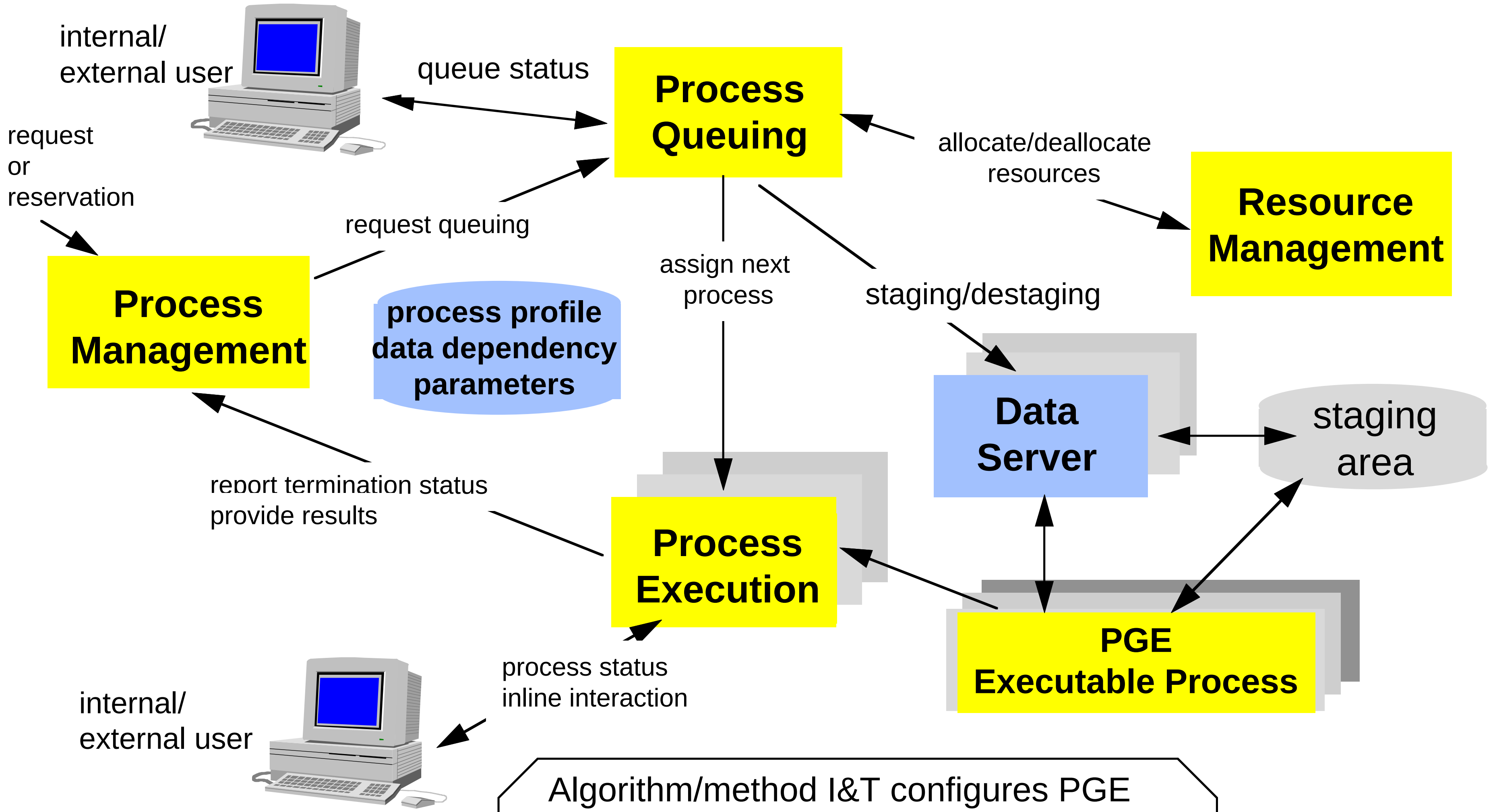
SDS - 4.5.7

# Processing Subsystem Context



SDS - 4.5.7

# Processing



internal/
external user

queue status

Process Queuing

request
or
reservation

request queuing

allocate/deallocate
resources

Resource Management

Process Management

process profile
data dependency
parameters

assign next
process

staging/destaging

Data Server

staging
area

report termination status
provide results

Process Execution

PGE
Executable Process

process status
inline interaction

internal/
external user

Algorithm/method I&T configures PGE and associated data required to manage execution

SDS - 4.5.7

# Processing

## Characteristics

- queuing service supports multiple queues - site definable
- queuing service feeds multiple executors - currently one per string
- queuing performs staging/destaging - includes predictive staging
- staging/destaging can be logical using data server services
- user can view queue and process status - also supports interaction with algorithm through executor
- I&T process supported through ‘client configuration’ to access OTS tools - prepares PGE for use in processing subsystem and establishes data used for management of processing [toolset will be site specific]

# Processing

## Outstanding Design Issues

- processing reservations - linked to Planning guaranteed service
- predictive staging approaches
- executable process access to Data Server - identify requirements

## Prototype Investigations

- using OTS developments to prototype queuing and process management services (e.g.  DQS)
- linked to distributed algorithm processing investigations and planning prototype to define service interfaces (utilizing pathfinder algorithms)

SDS - 4.5.7.3

# QA Support

Three general QA mechanisms supported - inline, offline, off-site (e.g SCF)

Considerable flexibility to meet various QA operational concepts

## Examples

- during I&T use inline QA to visually verify intermediate outputs before proceeding  with processing
- routine offline QA performed by DAAC staff - subscription on Data Server flags QA job when data is loaded into the Data Server.  Uses the functions of the Data Server to perform the QA procedure using specific Client configuration for this task.  Function can be instrument team supplied ‘data type service’.
- offsite QA performed by SCF team when certain conditions met (e.g.  >x% saturation).  Subscription on Data Server can check for this condition and notify relevant SCF team members with responsibility for this check. Investigation can proceed using the Data Server functionality at the DAAC, SCF functionality via machine-to-machine interfaces, or following transfer to SCF for local analysis